μWeather™ – An APRS-compatible weather station

David R. Andersen, K0RX
63 Pleasant Hill Road
Mt. Vernon, IA 52314

k0rx@arrl.net
http://www.qsl.net/k0rx/uWx

Abstract

In this paper, I describe an APRS-compatible, single-board weather station project that I have developed. The weather station is based on the Microchip 16F877 microcontroller. The basic weather station monitors temperature, relative humidity, and barometric pressure. These data are periodically formed into APRS weather packets sent using on-board AFSK generation. Thus, no TNC is required to transmit the weather packets over the air. In addition, the data are sent in industry-standard "datalogger" format out the serial port for use with any of several weather station monitoring programs. Future plans call for implementing the ability to upgrade the weather station by attaching an optional rain gauge and/or anemometer. Any rain gauge that emits a TTL pulse for each 0.01 inch of rain received as well as the Dallas Semiconductor 1-wire anemometer will work with μWeather™.

Introduction

In the last few years, the Automatic Packet Reporting System (APRS) has become one of the most popular amateur packet radio applications in existence. APRS links together a wide variety of different telemetry-producing devices such as weather stations, GPS trackers, and balloon and model rocket telemetry systems into a tremendously interesting and dynamic network of amateur radio enthusiasts. The public service applications of this network are widespread – helping to track everything from storm spotters to parades and automobile rallys, and providing a comprehensive set of weather data that is uplinked to a web page for retrieval by all interested parties are just a sample of the services that APRS provides. Recently, the National Weather Service (NWS) has begun to make use of this data, incorporating it into a NEMOnet page on the web that provides a comprehensive initial data set for weather prediction models used by the NWS to issue the forecasts that we all depend on. These public-service applications are precisely what makes APRS interesting and fun for the legion of ham radio operators that participate in this aspect of our diverse hobby.

We have all had the experience of watching the weather forecast on the television news report, and waking up the next morning to a completely different weather picture than what was predicted the night before. Such a situation can be frustrating at best, and dangerous at worst. In fact, I find the weather-reporting capabilities of APRS aid me in my own predictive efforts – and believe that I can do at least as good of a job as the pseudo-professionals on TV. However, there are difficulties with the weather data supplied by the APRS network. These difficulties arise because of two basic characteristics of commercial APRS-compatible weather stations: they are expensive and complex. Currently, a simple weather station reporting temperature, barometric pressure, and humidity (in my opinion the three most useful sets of predictive weather data) requires a basic station plus possibly additional sensors, a computer or other conversion device, and a TNC. All told, the costs of a basic weather station can easily climb into the multi-hundred dollar range, and that doesn't include the radio that actually sends the packets. As a result of the cost and complexity of these stations, it is simply not possible to deploy as many simple, reliable weather stations as is required for accurate forecasting.

Will Beals, N0XGA, and Russ Chadwick, KB0TVJ, have developed an excellent experimenters weather station kit for APRS use, the T238. This weather station is available through TAPR, and the necessary sensors, originally developed by Dallas Semiconductor, Inc. can be obtained quite easily as well. The T238 project is a big step in the right direction, with a significantly lower cost than commercially-available weather stations of comparable accuracy. However, this kit does not support the on-board generation of AFSK tones, and as a result an external TNC is required for APRS applications. The fact that all sensors are off-board, coupled with the extra expense and complexity required with the use of an external TNC makes this weather station less suitable for remote applications.

My goal in designing μWeather™ was to develop a weather station that would fulfill the need that I identified above. The station had to be simple, low-cost, and capable of accurately reporting weather data in a wide variety of deployment scenarios. I had been following the work of John Hansen, W2FS (PIC-based KISS TNC, http://john.hansen.net) and Byon Garrabrant, N6BG (TinyTrak, http://www.byonics.com/tinytrak), and thought that techniques similar to ones they had used for their APRS/Packet Radio projects could be used in a low-cost microcontroller-based weather station. The weather station described in this paper is the result of my development efforts, as well as able assistance from beta-tester Tony Arnerich, KD7TA.

Weather Station Concept

Initially, the concept for my weather station was quite simple – I wanted to be able to monitor temperature, barometric pressure, and humidity. To do this, the weather station would need to monitor the sensors and then generate AX.25 packets and the associated audio-frequency shift keying tones to send them. However, as in some of J. R. R. Tolkien's stories, this project grew in the development process. A serial port would be needed for the initial configuration of the weather station, including the station's callsign, a list of digipeaters, APRS parameters such as the station's location (latitude and longitude) and various transmission parameters such as the TXD setting. At some point during the conceptualization phase of the project, it occurred to me that it would be pretty straightforward to add serial datalogger output. After all, I already had the data,

and it was just a matter of formatting it properly. In addition, after conversations with a couple of friends of mine about the feature list for this station, I decided to add an LCD display, and rain gauge and anemometer capability as well. Thus, the basic station quickly transformed itself into something resembling a full-featured weather station. However, I've worked hard to make sure that the original design goal of creating a single-board weather station that is capable of monitoring very basic weather data and transmitting that over the APRS network using the AX.25 packet radio protocol has survived. The LCD display, datalogger monitoring, rain gauge, and anemometer are all options that do not have to be incorporated.

With the dust settling on the design phase of this project, μWeather™ has the following feature list:

- Single-board weather station microcontroller-based weather station
- On-board or remotable temperature/humidity/bar. pressure sensors
- Real-time clock for time-stamping APRS weather packets
- On-board generation of AFSK for APRS weather packets (no TNC required) with standard 5-pin DIN receptacle for interfacing with radio
- Weather data is also output in RS232 industry-standard datalogger format for compatibility with 3rd-party weather display software
- 20x4 character LCD for continuous display of current weather conditions
- Easy initial configuration performed through the μWeather™'s RS232 port with any standard terminal or terminal emulator software
- Future enhancements - inputs for optional rain gauge and anemometer

Design Description

The first step in implementing this project was to determine what microcontroller I was going to use. When I began the project, I had very little experience working with any microcontroller, so the field was pretty open. After examining the options, I settled on the Microchip 16F877 for three basic reasons: 1) this microcontroller has on-chip analog to digital conversion capability which would help reduce the chip count and therefore the cost, 2) it is flash programmable which would reduce the development time, and 3) it is based on the concept of reduced instruction set computing (RISC) which

simplified the coding process. I wanted to write the firmware in assembly language so this third item was a key consideration in my mind.

In order to achieve the specified design functionality, I needed to incorporate three sensors on the weather station board. These were temperature, barometric pressure, and relative humidity sensors. In addition, since weather data without a time-stamp is of little value, a real-time clock would be required. For temperature measurement, I chose to use the DS1621 digital thermostat, an 8-pin I^2C device. As a result, I opted for the 8-pin DS1307 real-time clock in order to use the same I^2C bus that I had already incorporated in the design. For the measurement of barometric pressure, I chose the MPX4115A sensor by Motorola, and for relative humidity measurement, I used the HIH-3610. Both of these last two sensors put out a voltage that is proportional to the parameter they are measuring, and so are read using the on-chip A/D capability of the 16F877.

The most difficult of these three sensors to incorporate in the design was the barometric pressure module. The 16F877's on-board A/D has 10-bit resolution, which means that for A/D voltage references of 0 and 5V respectively, the MPX4115A could be read with a resolution of about 1.1 mBar. This was an order of magnitude poorer resolution than the APRS specification permits. However, the 16F877 has a facility for using voltage references other than the power supply voltages in conjunction with it's A/D module. Thus, through the use of judicious hardware design, plus digital filtering techniques in the firmware, it was possible to enhance the barometric pressure resolution to approximately 0.1 mBar: exactly what the APRS specification permits.

On-board constant-phase AFSK tones are generated using a 4-bit resistive ladder D/A converter. This approach uses a standard technique that is well-documented in Microchip application note AN655, and waveforms generated in this manner were used successfully in the TinyTrak APRS tracker by N6BG. I generate my AFSK tones using delay tables, and the bit period is fixed using the TMR0 interrupt facility of the 16F877.

Configuration data is input to and serial datalogger data is output from the 16F877 through a bit-bang RS232 serial port in 2400 baud 8-N-1 format. I chose to use my bit-bang routines, rather than the built-in UART on the 16F877 primarily because my early development work had been done on a 16F84 microcontroller that did not have an on-

board UART. However, the the TX and RX pins for the built-in serial port have not been used elsewhere in the weather station design in order to permit the use of a simple boot-loader for firmware upgrades in the future. The 2400 baud rate was chosen in order to be compatible with the industry-standard datalogger output format. In order to maintain the design philosophy and keep the chip count down, I have not used a level-converter, and the voltage levels are TTL. This may give rise to compatibility issues with some computer serial ports, however this has not been a problem to date.

I selected an industry-standard 4x20 HD44780-based LCD text module for the display. This permits me to output current weather data in a user-readable format with a relatively low cost. Such display modules are available surplus for very little money. In addition, the LCD interface is designed such that removing the LCD for remote operation will not affect the functionality of the weather station. One of my future plans for enhancement is to substitute an LCD graphics display for the text module so that I can plot historical weather data as well. Such a modification would greatly enhance the utility of the weather station for stand-alone weather prediction efforts.

Currently, approximately 4K of the 8K available 16F877 program space has been used by the μWeather™ firmware. This leaves plenty of room for enhancements such as the graphics display and the rain gauge/anemometer. Also, μWeather™'s current draw is less than 15 mA with the LCD attached, indicating that battery/solar-powered operation is feasible.

Summary

In summary, this paper describes the development of a low-cost, APRS-compatible, single board weather station suitable for remote deployment. The station is based on a Microchip 16F877 flash microcontroller. The station monitors and periodically formats temperature, barometric pressure, and humidity data into APRS weather packets and transmits them using AFSK tones generated on-board. The station also outputs the weather data in industry-standard serial datalogger format. Up-to-date information concerning μWeather™'s development status may be obtained at the project's web page: http://www.qsl.net/k0rx/uWx.

Acknowledgements

1) μWeather™ is a trademark of David R. Andersen, K0RX.
2) APRS is a trademark of Bob Bruninga, WB4APR.
3) The hard work of this project's beta tester, Tony Arnerich, KD7TA is greatfully acknowledged. Without his efforts, the project would be much less than it is.